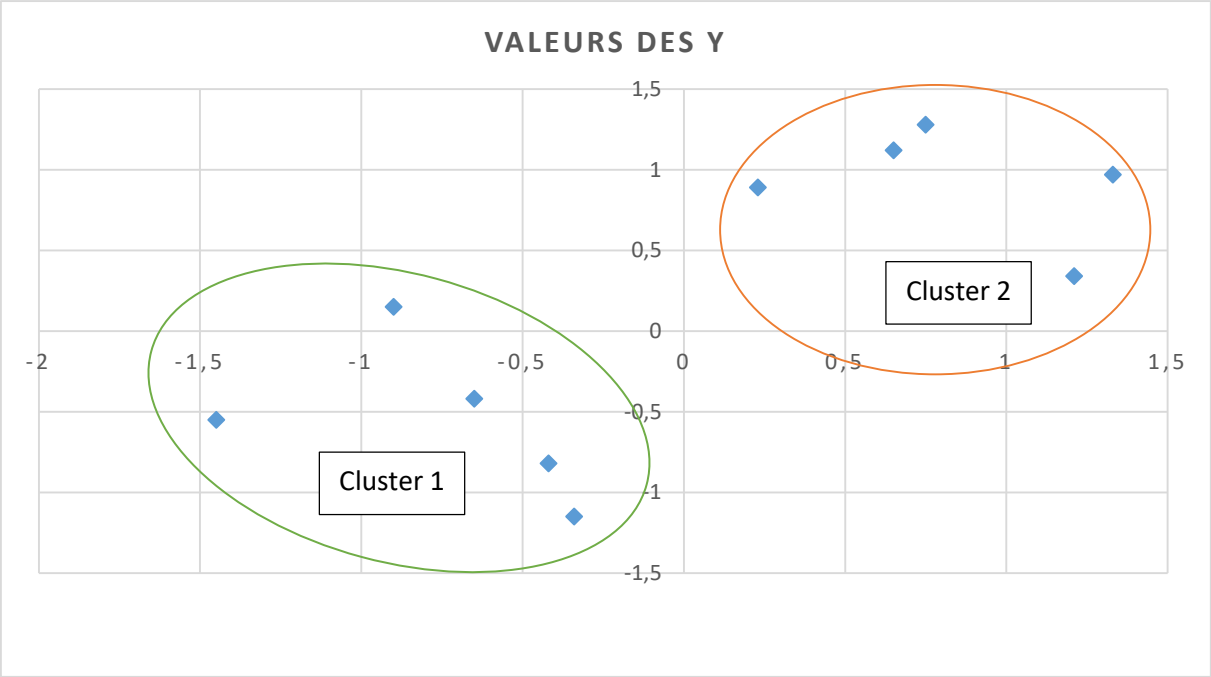
TP4 : PARALLELISATION D'UN K-NN

Objectif :

Dans ce TP nous allons implémenter un algorithme appelé K-NN (en anglais K-Nearest Neighbours). Il s'agit d'un algorithme d'apprentissage supervisé qui se base sur la méthode de recherche des K voisins les plus proches. Cet algorithme, utilisé principalement dans des tâches de classification, repose sur l'idée que les objets similaires sont souvent les plus proches les uns des autres dans l'espace des données. Il utilise la distance euclidienne pour le calcul de la distance entre les points et K comme paramètre définissant le nombre de voisins à prendre en considération pour la prise de décision.

La classe d'une donnée est prédite en fonction des données les plus proches en se basant sur des caractéristiques spécifiques au domaine des données(Voir illustration ci-dessous).



Voici les étapes de l'algorithme K-NN :

- En entrée : Un ensemble de données, un nombre entier K
- Pour un nouveau point dont on veut prédire son cluster d'appartenance :
    1. Calculer toutes les distances de point avec les tous les autres points
    2. Retenir les K valeurs du jeu de données les plus proches de ce point
    3. Choisir la valeur dominante parmi les K valeurs précédentes
    4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour ce point.

TP4 : PARALLELISATION D'UN K-NN

Implémentation séquentielle :

1. *Définition des points et clusters:*

Nous commençons par la définition des classes :

- Cluster
- Point

Compléter le code fourni sur moodle pour ce faire.

2. *Les fonctions « helper » :*

Dans cet exercice nous allons implémenter les fonctions d'aide suivantes :

- ✓ Initiation d'un espace de données générées aléatoirement
- ✓ Calcul de la distance Euclidienne entre deux points
- ✓ Prédiction basée sur la distance euclidienne afin de définir un cluster
- ✓ Mise à jour des clusters et des points

3. *Implémentation séquentielle :*

Implémenter dans le « main », à l'aide de 1 et 2 l'algorithme séquentiel de la classification K-NN. Cet algorithme prend en entrée un points aléatoire et prédit le groupe/cluster auquel ce point appartient.

- ✓ Mesurer le temps d'exécution total
- ✓ A 'aide de plusieurs points de mesures de temps, quel est la partie de l'algorithme qui prends le plus de temps ?
- ✓ Qu'en déduisez-vous ?

4. *Implémentation parallèle :*

Dans cette question vous allez paralléliser votre programme à l'aide d'Open-MP.

- ✓ Instrumenter au bon endroit votre programme séquentiel, à l'aide de : #pragma omp
- ✓ Quel le nouveau temps d'exécution
- ✓ Peut-on faire mieux ?

5. *Génération des graphiques*

Afin de rendre visuel vos calculs, ajouter la génération de graphes d'entrée et de sortie de votre algorithme. Utiliser différente couleurs pour les différents clusters pour améliorer la lecture des graphes.